实验9 Gesture、UIAlertController、 ScrollView

（综合性实验 2学时）

1、实验目的

• 理解并掌握iOS多点手势识别的相关技术；

2、实验要求

• 认真填写实验报告，要求附加部分运行界面和主要代码；

• 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；

3、实验内容

• Gesture

• 分别采用代码随机位置大小生成和直接拖拽的方式产生多个视图；

*//MARK*：随机添加*UILabel*

**@IBAction** **func** add(**\_** sender: UIButton) {

*//*随机设置原点位置

**let** rect = **self**.view.bounds

**let** x = Int(arc4random()) % Int(rect.size.width)

**let** y = Int(arc4random()) % Int(rect.size.height)

*//*随机设置颜色

**let** red = CGFloat(arc4random()) / CGFloat(UInt32.max)

**let** green = CGFloat(arc4random()) / CGFloat(UInt32.max)

**let** blue = CGFloat(arc4random()) / CGFloat(UInt32.max)

**let** color = UIColor(red: red, green: green, blue: blue, alpha: 1)

**let** addLabel = UILabel(frame: CGRect(x: x, y: y, width: 30, height: 30))

addLabel.backgroundColor = color

**self**.view.insertSubview(addLabel, at: 0)

*//*为添加的*Label*添加手势

**let** panGes = UIPanGestureRecognizer(target: addLabel, action: **#selector**(pan))

addLabel.addGestureRecognizer(panGes)

**let** tapGesture = UITapGestureRecognizer(target: addLabel, action: **#selector**(tapTwo))

tapGesture.numberOfTapsRequired = 2

addLabel.addGestureRecognizer(tapGesture)

*//*重要：设置后才能使用手势

addLabel.isUserInteractionEnabled = **true**

}

• 采用简单的动画进行移动；

*//MARK:*随机移动*UILabel*

**@IBAction** **func** move(**\_** sender: UIButton) {

**for** v **in** **self**.view.subviews{

**if** v **is** UILabel{

**let** rect = **self**.view.bounds

**let** x = Int(arc4random()) % Int(rect.width)

**let** y = Int(arc4random()) % Int(rect.height)

UIView.animate(withDuration: 2, animations: {

v.center = CGPoint(x: x, y: y)

}, completion: **nil**)

}

}

}

• 给视图加上阴影(layer)；

*//MARK:*设置圆角

moveView.layer.cornerRadius = 20

*//MARK:*设置阴影

moveView.layer.shadowColor = UIColor.black.cgColor

moveView.layer.shadowOffset = CGSize(width: 5, height: 5)

moveView.layer.shadowOpacity = 0.5

• 可全部清空子视图；

*//MARK:*全部删除*UILabel*

**@IBAction** **func** clear(**\_** sender: UIButton) {

**for** v **in** **self**.view.subviews{

**if** v **is** UILabel{

v.removeFromSuperview()

}

}

}

• 视图支持手势（pan移动、tap删除、pinch缩放、rotation旋转）；

提示：Pinch的scale属性可用于调整frame

rotation需要用transform属性实现

*//MARK*：手势设置

*//*拖动识别器设置

**let** panGesture = UIPanGestureRecognizer(target: **self**, action: **#selector**(pan))

moveView.addGestureRecognizer(panGesture)

*//*放大，缩小识别器设置

**let** pinchGesture = UIPinchGestureRecognizer(target: **self**, action: **#selector**(pinch))

moveView.addGestureRecognizer(pinchGesture)

*//*旋转识别器

**let** rotationGesture = UIRotationGestureRecognizer(target: **self**, action: **#selector**(rotation))

moveView.addGestureRecognizer(rotationGesture)

*//*点击识别器

**let** tapGesture = UITapGestureRecognizer(target: **self**, action: **#selector**(tapTwo))

tapGesture.numberOfTapsRequired = 2

moveView.addGestureRecognizer(tapGesture)

}

*//*拖动方法实现

**@objc** **func** pan(recognizer: UIPanGestureRecognizer){

**switch** recognizer.state{

**case** .changed, .ended:

**let** moveDisPoint = recognizer.translation(in: **self**.view) *//*返回想*x*，*y*移动的距离，*CGPoint*类型

moveView.center.x += moveDisPoint.x

moveView.center.y += moveDisPoint.y

recognizer.setTranslation(CGPoint.zero, in: **self**.view) *//*重置*translation*（速度重置）

**default**:

**return**

}

}

*//*放大*,*缩小方法实现

**@objc** **func** pinch(recognizer: UIPinchGestureRecognizer){

**switch** recognizer.state {

**case** .changed, .ended:

**let** scale = recognizer.scale *//*返回放大，缩小的倍数

moveView.bounds.size = CGSize(width: moveView.bounds.width \* scale, height: moveView.bounds.height \* scale)

recognizer.scale = 1 *//*重置

**default**:

**return**

}

}

*//*旋转方法实现

**@objc** **func** rotation(recognizer: UIRotationGestureRecognizer){

**switch** recognizer.state {

**case** .changed, .ended:

**let** rotation = recognizer.rotation *//*返回旋转度数

moveView.transform = CGAffineTransform(rotationAngle: rotation)

**default**:

**return**

}

}

*//*双击方法实现*(*消失*)*

**@objc** **func** tapTwo(recognizer: UITapGestureRecognizer){

**switch** recognizer.state {

**case** .recognized:

moveView.removeFromSuperview()

**default**:

**return**

}

}

• 实现UIAlertController交互

• 显示ActionSheet并进行交互；

*//MARK:*点击按钮显示下方*actionSheet*框

**@IBAction** **func** actionSheetShow(**\_** sender: UIButton) {

**let** actionSheet = UIAlertController(title: "菜单栏", message: "选择你喜欢的选项", preferredStyle: .actionSheet)

**let** action1 = UIAlertAction(title: "选项一：吃饭.destructive", style: .destructive, handler: {

(alertAction) **in**

print("吃饭")

})

**let** action2 = UIAlertAction(title: "选项三：吃饭和睡觉.cancel", style: .cancel, handler: {

(alertAction) **in**

print("吃饭和睡觉")

})

**let** action3 = UIAlertAction(title: "选项二：睡觉.default", style: .default, handler: {

(alertAction) **in**

print("睡觉")

})

actionSheet.addAction(action1)

actionSheet.addAction(action2)

actionSheet.addAction(action3)

present(actionSheet, animated: **true**, completion: **nil**)

}

• 显示Login Alert并进行交互；

*//MARK:*点击按钮显示中间*alert*

**@IBAction** **func** alertShow(**\_** sender: UIButton) {

**let** alert = UIAlertController(title: "Login Alert", message: "Please input your username and password", preferredStyle: .alert)

alert.addTextField { (textField) **in**

textField.placeholder = "请输入你的用户名"

}

alert.addTextField(configurationHandler: {

(textField) **in**

textField.placeholder = "请输入你的密码"

textField.isSecureTextEntry = **true** *//*设置*UItextField*为密码格式

})

**let** action1 = UIAlertAction(title: "取消", style: .cancel, handler: **nil**)

**let** action2 = UIAlertAction(title: "确认", style: .default) { (UIAlertAction) **in**

*//*点击确认后执行

**if** **let** username = alert.textFields?.first?.text, **let** password = alert.textFields?.last?.text{

print("你的用户名为\(username)")

print("你的密码为\(password)")

}

}

alert.addAction(action1)

alert.addAction(action2)

*//*设置*UIAlertController*显示

**self**.present(alert, animated: **true**, completion: **nil**)

}

• 一个界面使用两个scrollView

• 在一个scrollView中可进行多张图片横屏滚动浏览(相册)，需要有pagecontrol进行提示；

*//scrollView2*添加多张图片

**let** total = 4

addAllImage(total: total)

*//pageControl*设置

pageControl.numberOfPages = total

pageControl.currentPage = 0

*//scrollView2*添加*4*张图片

**func** addAllImage(total:Int){

**for** i **in** 1...total{

**let** imageView = UIImageView(frame: CGRect(x: Int(scrollView2.bounds.width) \* (i - 1), y: 0, width: Int(scrollView2.bounds.width), height: Int(scrollView2.bounds.height)))

imageView.image = UIImage(named: "\(i)")

scrollView2.addSubview(imageView)

}

scrollView2.contentSize = CGSize(width: Int(scrollView2.bounds.width) \* total, height: Int(scrollView2.bounds.height))

scrollView2.isPagingEnabled = **true** *//*设置分页，显示整张图片

scrollView2.showsHorizontalScrollIndicator = **false** *//*隐藏垂直滚动条

}

*//(*减速结束后*)--*设置圆点改变

**func** scrollViewDidEndDecelerating(**\_** scrollView: UIScrollView) {

**let** current = scrollView2.contentOffset.x / scrollView2.bounds.width

pageControl.currentPage = Int(current)

}

*//*点击*pageControl*圆点切换图片

**@IBAction** **func** pageChage(**\_** sender: UIPageControl) {

**let** current = pageControl.currentPage

scrollView2.scrollRectToVisible(CGRect(x: Int(scrollView2.bounds.width) \* current, y: 0, width: Int(scrollView2.bounds.width), height: Int(scrollView2.bounds.height)), animated: **true**)

}

• 在另一个scrollView中可放大缩小；

提示：需用delegate

*//scrollView*添加图片*(*使其可以缩放*)*

**let** imageView = UIImageView(frame: CGRect(origin: CGPoint(x: 0, y: 0), size: CGSize(width: 600, height: 400)))

imageView.image = UIImage(named: "3")

scrollView.addSubview(imageView)

scrollView.contentSize = imageView.bounds.size *//*重要，设置*scrollView*内容*size*，才能拖动

scrollView.minimumZoomScale = 0.5 *//*设置缩小最小尺寸

scrollView.maximumZoomScale = 2.0 *//*设置放大最大尺寸

*//(*缩放的视图*)*

**func** viewForZooming(in scrollView: UIScrollView) -> UIView? {

**return** scrollView.subviews.first

}